

## Прикладная математика.

Наступивший век - это век всеобщей информатизации. Математические расчеты, проводимые с помощью компьютеров, прочно проникли в самые разнообразные научные дисциплины, технику, экономику, управление и другие сферы человеческой деятельности, причем не только в традиционные области приложения математики, но и туда, где математика используется совсем недавно.

Применяемый математический аппарат стал намного разнообразнее. Если еще 30 – 50 лет назад инженерам требовались знания, лежащие в пределах школьной математики, дифференциального и интегрального исчисления, то теперь применяется аппарат ранее неизвестный и даже целые математические дисциплины, например линейное программирование, созданные в первую очередь с целью приложения.

В результате повысились требования к математической подготовке инженера. Это связано с все большим распространением компьютерного (модельного) подхода к решению инженерных задач. Понятие «моделирование» стало за последнее десятилетие едва ли не самым распространенным в естественно - научной и технической литературе. Сегодня трудно представить проектную или конструкторскую организацию, не использующую моделирования в своей практике в той или иной мере. И чем сложнее проектируемый объект, тем важнее роль моделирования в его изучении и создании.

Нет никакого сомнения в том, что развитие и применение математических моделей и математического аппарата будут только еще более усиливаться. Этим объясняется возросший интерес к вопросу о том, как создаются математические модели, как они изучаются и как интерпретируются.

**Моделирование** – один из основных методов познания, является формой отражения действительности и заключается в выяснении или воспроизведении тех или иных свойств реальных объектов предметов и явлений с помощью других объектов, процессов, явлений, либо в виде изображения, совокупности уравнений, алгоритмов и программ.

**Моделью** называется объект-заместитель, который в определенных условиях может заменять объект-оригинал, воспроизводя интересующие нас свойства и характеристики оригинала, причем имеет существенные преимущества:

- дешевизну;
- наглядность;
- легкость оперирования и т.п.

**Модель** – образ реальной системы (объекта, процесса) в материальной или теоретической форме. Этот образ отражает существенные свойства объекта, он замещает реальный объект в ходе исследования и управления.

**Модель** отображает (описывает, воспроизводит) некоторые интересующие черты исследуемого объекта.

**Типы моделей: физические или аналоговые** (модели самолетов в аэродинамической трубе, план, карта, график) и **математические** модели, где количественно определенные переменные связаны математическими законами ( модели в физике, экономике).

Своего рода графической моделью является проект землеустройства, топологическая карта, схема и пр.

**Математическая модель**- это приближенное описание какого-либо класса явлений внешнего мира, выраженное с помощью математической символики, то есть система математических соотношений, приближенно в абстрактной форме описывающих изучаемый процесс или систему.

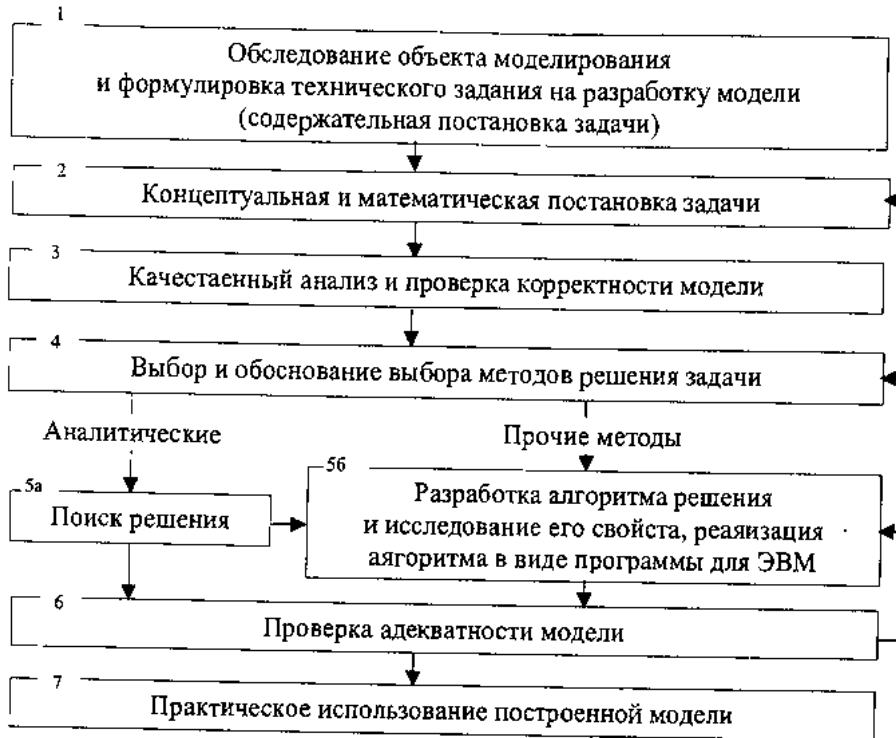
### **Создание математической модели.**

При построении модели реальное явление неизбежно упрощается, схематизируется, и эта схема описывается с помощью того или другого математического аппарата. Чем удачнее будет подобрана математическая модель, чем лучше она будет отражать характер-

ные черты явления, тем успешнее будет исследование и полезнее – вытекающие из него рекомендации.

Общих способов построения математических моделей не существует. В каждом конкретном случае модель выбирается исходя из задачи исследования (какие параметры требуется определить и влияние каких факторов отразить). Математическая модель должна отражать важнейшие черты явления и вместе с тем должна быть по возможности простой.

#### Основные этапы создания математической модели.



При создании модели встает вопрос о решении математической задачи. В какой бы математической форме не была реализована модель в дифференциальной, линейной или другой, необходимо найти решение. В качестве компонентов математических моделей часто встречаются такие как решение различного вида уравнений и их систем, а также вычисление интегралов. И далеко не все эти математические задачи можно решить аналитически. Здесь и вступает в свои права **прикладная математика**.

Раздел математики, имеющий дело с созданием и обоснованием численных алгоритмов для решения сложных задач различных областей науки, называют **прикладной математикой**.

**Главная задача прикладной (вычислительной) математики** – фактическое нахождение решения с требуемой (оцениваемой) точностью. Этим она отличается от классической математики, которая основное внимание уделяет исследованию условий существования и свойств решения.

В истории прикладной математики можно условно выделить 3 периода.

Первый начался 3 – 5 тысячи лет назад. Он был связан с ведением конторских книг, вычислением площадей и объемов, расчетами простейших механизмов; иными словами – с несложными задачами арифметики, алгебры и геометрии. Вычислительными средствами служили сначала собственные пальцы, а затем – счеты. Исходные данные содержали мало цифр, и большинство выкладок выполнялось точно, без округлений.

Второй период начался с Ньютона. В этот период решались задачи астрономии, геодезии и расчета механических конструкций, сводящиеся либо к обыкновенным дифференциальным уравнениям, либо к алгебраическим системам с большим числом неизвестных. Вычисления выполнялись с округлением; нередко от результата требовалась высокая точность, так что приходилось сохранять до 8 значащих цифр.

Вычислительные средства стали разнообразнее: таблицы элементарных функций, затем арифмометр и логарифмическая линейка. Скорость этих средств была невелика, и вычисления занимали дни, недели и даже месяцы.

Третий период начался примерно в 40-х годах 20 века. Военные задачи требовали недоступных человеку скоростей и привели к разработке ЭВМ. Скорость даже простейших ЭВМ настолько превосходила скорость механических средств, что стало возможным проводить вычисления огромного объема. Это позволило численно решать новые классы задач. Для решения многих задач до сих пор используются методы разработанные в «доэлектронный» период, но также появилось множество новых численных методов.

**Целью изучения данной дисциплины являются:**

- Знакомство с принципами построения алгоритмов и методикой постановки задач для приближенного решения на ЭВМ.
- Изучение вычислительных методов, применяемых при решении прикладных задач, не имеющих аналитического решения, либо имеющих его, но, по ряду причин, получение которого затруднено;
- Ознакомление с основными источниками погрешностей, их оценкой и методами устранения;

Сложные вычислительные задачи, которые необходимы для получения решения математических моделей, можно разбить на ряд элементарных – таких как вычисление интеграла, решение дифференциального уравнения, решение нелинейных уравнений и систем. Часть из них может быть решена аналитически, однако для большинства используются приближенные вычислительные методы. В курсе высшей математики мы уже познакомились с таким приближенным методом как разложение функций в степенные ряды Тейлора.

Распространенным способом решения является применение численных алгоритмов (численных методов). Их существует великое множество и первоначально они не были связаны с применением ЭВМ, а осуществлялись также вручную. Появление компьютеров сделало очень удобным и простым (поскольку они используются и в пакетах математических программ) их применение. В настоящее время самые распространенные методы записаны в книгах уже в виде готовых программ, «численных рецептов». Мы воспользуемся этими рецептами для некоторых распространенных задач: решения нелинейных уравнений, систем линейных уравнений, вычисления определенного интеграла, решения дифференциальных уравнений, методам численной аппроксимации функций.

#### **Макросы VBA Excel и их базовые элементы.**

Следует сказать, что разработанный численный метод содержит обычно только принципиальную схему решения задачи, не включая детали, без которых невозможна реализация метода на ЭВМ. Необходимо подробно детализировать все этапы вычислений, то есть составить программу на выбранном языке программирования, например Фортран, Си и пр. Мы будем реализовать решение вычислительных задач в виде макросов на языке VBA (Visual Basic for Application).

Программа представляется в виде последовательности строк, описывающих процесс (алгоритм) решения поставленной задачи. Каждая из программных строк состоит из одного или нескольких операторов.

Макросом (или макрокомандой) в EXCEL называется последовательность операций, снабженная именем, которая может быть выполнена автоматически. Программа, написанная на VBA EXCEL – макрос. Кроме этого можно называть такую программу приложением или проектом.

Для того чтобы пользоваться программами (макросами) не обязательно быть программистом, тем более что при изучении дисциплины будут использоваться готовые программы. Однако, даже при использовании готовых макросов, надо понимать, что записано в тексте кода(текста программы), какая последовательность действий будет осуществляться, в каких строках и почему необходимо сделать изменения. Поэтому мы познакомимся с базовыми конструкциями, имеющимися во всех алгоритмических языках – операторами

присваивания, условными операторами, циклами, массивами и подпрограммами, с тем, чтобы была возможность модифицировать макросы для конкретных нужд.

### Операторы ввода – вывода, присваивания.

Пример простейшей программы:

```
Sub Пифагор()  
a = 3  
b = 4  
c = Sqr(a ^ 2 + b ^ 2)  
End Sub
```

Первая строка кода заглавие программы, с этой строки начинается любая программа. Здесь содержится имя программы, в данном случае **Пифагор()**. Любая программа заканчивается оператором End Sub.

В блок – схеме алгоритма этим двум строкам соответствует обозначение:



Внутри этих двух строк операторы программы. Мы записали 3 оператора которые называются операторы присваивания. Общий вид:  $a = b$ . Справа от равенства записывается алгебраическое выражение, вычисленное значение которого будет присвоено переменной стоящей слева. Арифметические действия в выражениях совпадают с аналогичными операциями в EXCEL.

Математическая запись одного из операторов  $c = \sqrt{a^2 + b^2}$ .  $a, b, c$  – переменные. Переменные в программе имеют такой же смысл, как и в математике. Перед тем как использовать переменную ее рекомендуется описать (объявить).

Синтаксис оператора описания переменной:

**Dim** переменная [As тип].

Здесь **Dim** – ключевое слово описывающее переменную. (сокращение от слова dimension – размер). Тип данных может быть целым (Integer), вещественным (Single), текстом (String) и пр. В простейших программах строго описывать необходимо только массивы.

Кроме математических действий в операторе присваивания могут содержаться математические функции. Например:  $c = \text{Sqr}(a^2 + b^2)$  эквивалентно математической записи  $c = \sqrt{a^2 + b^2}$ . Некоторые математические функции, используемые в программах:  $\sin(x)$ ,  $\cos(x)$ ,  $\text{sqr}(x)$  ( $\sqrt{x}$ ),  $\text{exp}(x)$  ( $e^x$ ),  $\log(x)$  ( $\ln(x)$ ),  $\text{abs}(x)$  ( $|x|$ ),  $\text{atn}(x)$  ( $\arctg(x)$ ).

Часть математических функций в VBA отсутствует и их необходимо рассчитывать по формулам:

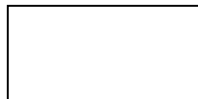
$$\arcsin(x) = \text{atn}(x / \text{sqr}(1 - x * x))$$

$$\arccos(x) = \frac{\pi}{2} - \text{atn}(x / \text{sqr}(1 - x * x))$$

$$\text{ch}(x) = (\text{exp}(x) + \text{exp}(-x)) / 2$$

$$\text{sh}(x) = (\text{exp}(x) - \text{exp}(-x)) / 2$$

Исполняемые операторы в блок – схемах обозначаются:



В примере простейшей программы двум переменным  $a$  и  $b$  присваиваются конкретные значения. Чтобы рассчитать переменную  $c$  уже с другими значениями нужно внести изменения в программу. Это неудобно. Действия в программе должны выполняться многократно с разными значениями входных данных. Поэтому значения переменных  $a$  и  $b$  вводятся в программу с листа MS EXCEL.

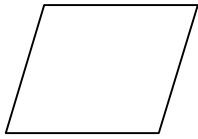
a = Range("a").Value

b = Range("b").Value

Вывод результатов осуществляется на лист MS EXCEL.

Range("c6").Value = c

В блок – схемах алгоритмов операции ввода – вывода соответствует диаграмма:



### Условные операторы и операторы перехода.

Операторы рассмотренной программы выполнялись поочередно. Это линейная программа. Для изменения последовательности выполнения операторов (т.е. для ветвления) программы используются операторы перехода.

Оператор безусловного перехода:

GOTO метка («номер строки»)

Если в определенном месте программы необходимо выполнить те или иные операторы в зависимости от некоторых условий, то применяются операторы условного перехода.

IF условие THEN оператор

Под условием понимается некоторое логическое выражение (высказывание), которое может быть либо истинным (TRUE – логическая 1), либо ложным (FALSE – логический 0). Условие часто записывается в виде арифметического отношения с использованием конструкций “>”, “<”, “>=”, “<=”, “< >” (≠). Если условие истинно, то оператор выполняется, если ложно, то выполняется следующий.

Пример:

IF (X > 12 Or X < 5) THEN X = X + 1

Если при истинности условия требуется выполнить несколько операторов, то используют конструкцию вида:

IF условие THEN

операторы

END IF

В случае, когда имеется альтернатива, то условный оператор приобретает вид:

IF условие THEN

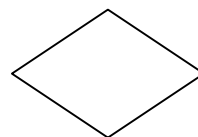
операторы 1

ELSE

операторы 2

END IF

В блок-схемах условные операторы обозначаются:



### Оператор цикла и пользовательские функции.

Для многократного выполнения блока операторов необходимо использовать оператор цикла. Когда количество выполнений заданного блока операторов известно заранее то используется оператор FOR...TO...STEP...NEXT.

Вид оператора:

FOR счетчик = начало TO конец [STEP шаг]

операторы

NEXT

Счетчик это переменная, которой присваивается значение *начало*. Затем значение этой переменной увеличивается на величину *шага* (по умолчанию шаг равен 1). Операторы цикла выполняются до тех пор, пока значение счетчика не достигнет конечного значения.

**Пример:**

```
For i = 1 To 10
  x = a + i * h
Next i
```

**Пользовательские функции** – это блок операторов, предназначенный для многократного выполнения в разных точках программы, оформляется как новая процедура (программа). Процедуры делятся на функции и подпрограммы.

Пользовательская функция:

```
FUNCTION название (параметры) [As тип]
  операторы
END FUNCTION
```

Пользовательская подпрограмма:

```
SUB название (параметры)
  операторы
END SUB
```

Обращение к подпрограмме: CALL *название (параметры)*

**Пример:**

```
Function fnf(x) As Single
  fnf = Sqr(2 * x + 1)
End Function
```

### Решение нелинейных уравнений.

**Постановка задачи:**

Найти коэффициент погрешности прибора  $\sigma$  при проведении геодезических измерений из уравнения:

$$\delta \cdot \cos \sigma - v \cdot \sigma^2 + \eta = 0$$

Значения  $\delta = 0,186$ ,  $v = 4,18$ ,  $\eta = 0,124$ .

Т.е. необходимо решить уравнение:

$$0,186 \cdot \cos \sigma - 4,18 \cdot \sigma^2 + 0,124 = 0$$

Это нелинейное уравнение, для которого нет аналитического метода. Для его решения можно применить численные методы математики.

**Общие понятия.**

Всякое уравнение с одним неизвестным можно записать в виде

$$f(x) = 0 \quad (1)$$

Корнем (или решением) уравнения (1) называется значение  $x$  такое, что  $f(x) = 0$ .

Уравнения делятся на алгебраические и трансцендентные (т.е. уравнения, содержащие тригонометрические функции или другие специальные функции типа  $e^x$ ,  $\ln x$  и т.д.).

Пример алгебраич. уравнения:  $x^n + a_{n-1}x^{n-1} + \dots + a_1x + a_0 = 0$ . (2)

Для уравнения (2) явные формулы найдены только до  $n = 4$ . Для уравнений более высоких степеней точных решений не существует (т. Абеля). Т.е. под задачей отыскания решений уравнения (1) будем понимать задачу вычисления с заданной точностью  $\epsilon$  конечного числа подлежащих определению корней этого уравнения (обычно задачу уточняют, т.е. какие именно корни нужно найти – положительные, максимальные, или корни из заданного интервала). Корни могут быть простые и кратные.

Методы решения прямые (с помощью формул, позволяющих найти точное решение) и итерационные (т.е. многократное повторение некоторого алгоритма с нахождением определенного приближенного решения).

Основные этапы решения:

- 1) Локализация (отделение) корней.
- 2) Уточнение решения (итерационное уточнение корня)

Первый этап – локализация корней.

**Определение.** Отрезок  $[a, b]$  содержащий только один корень  $x$  уравнения (1) называется отрезком локализации корня  $x$ .

Т.е. под отделением корня понимают нахождение отрезка, на котором лежит этот и только этот корень данного уравнения.

Способы отделения корней многообразны. Иногда из физ. сообр. графический метод. Построение таблиц функции. При этом о локализации корня судим по перемене знаков.

Несколько теорем из математического анализа.

**Теорема 1.** Если функция  $y = f(x)$  непрерывна на отрезке  $[a, b]$  и принимает на его концах значение разных знаков, т.е.  $f(a) \cdot f(b) < 0$ , то внутри отрезка  $[a, b]$  существует, по крайней мере, один корень уравнения (1).

**Теорема 2.** Если функция  $y = f(x)$  непрерывна на отрезке  $[a, b]$ ,  $f(a) \cdot f(b) < 0$  и  $f'(x)$  на интервале сохраняет знак, то внутри отрезка  $[a, b]$  существует единственный корень уравнения  $f(x) = 0$ .

Примеры локализации корней.

Графический:  $4(1 - x^2) - e^x = 0 \Rightarrow 1 - x^2 = 0,25 e^x$

Строим график: (рисунок)

Отрезки  $[-1, 0]$  и  $[0, 1]$  являются отрезками локализации.

**Пример.** Локализация корней уравнения

$$x^3 - 1,1x^2 - 2,2x + 1,8 = 0$$

с помощью таблицы на отрезке  $[-2, 2]$

$x$	-2	-1,6	-1,2	-0,8	-0,4	0	0,4	0,8	1,2	1,6	2
$f(x)$	-6,2	-1,592	1,128	2,344	2,44	1,8	0,808	-0,152	-0,696	-1,296	-2,000

Функция меняет знак на концах отрезков  $[-1,6; -1,2]$ ;  $[0,4; 0,8]$ ;  $[-1,6; 2,0]$

Второй этап.

**Итерационное уточнение корня.**

Если найдены отрезок  $[a, b]$  на котором есть один корень уравнения, то любую точку отрезка можно принимать за приближенное значение корня. При этом погрешность приближения не превышает длины  $[a, b]$ . Т.е. задача нахождения приближенного значения сводится к нахождению отрезка  $[a, b]$  ( $|b - a| < \varepsilon$ ) содержащий только один корень уравнения.

**1. Метод половинного деления (бисекции).**

За приближенное значение можно принять середину отрезка  $[a, b]$ .

$$c = (a+b)/2.$$

Проверяя знаки  $f(a), f(b), f(c)$  выясним в каком из отрезков  $[a, c]$  или  $[c, b]$  содержится корень

$$x^* \in [a, c], \text{ если } f(a)f(c) < 0;$$

$$x^* \in [c, b], \text{ если } f(c)f(b) < 0.$$

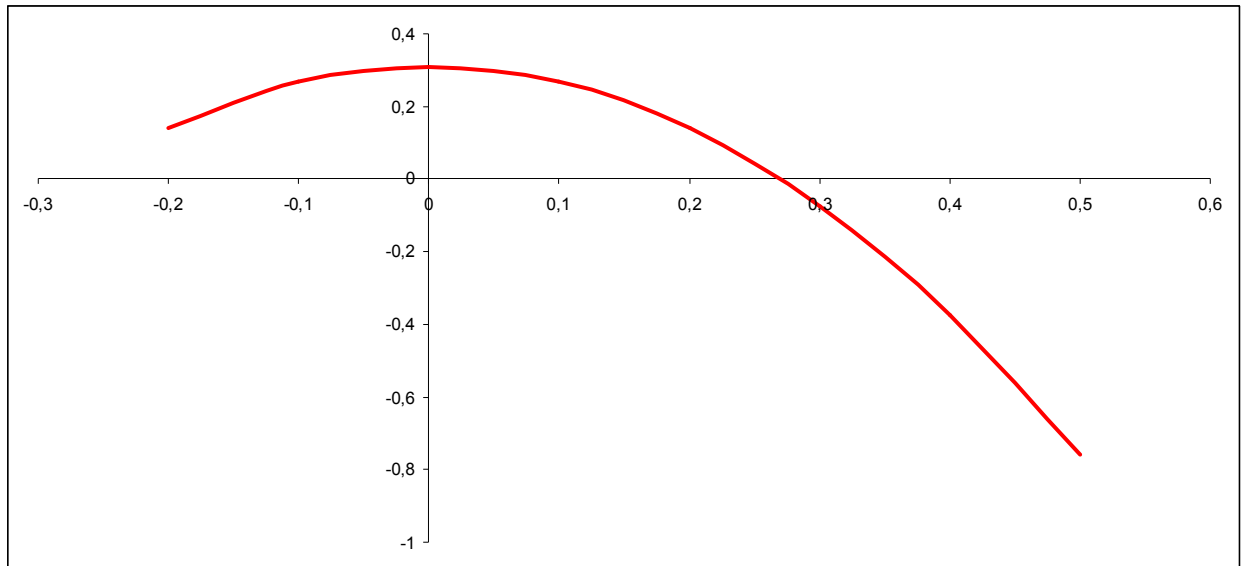
Выбранный отрезок принимаем за  $[a, b]$  и повторяем это до тех пор, пока получаемый отрезок не сожмется до заданной степени точности ( $|b - a| < \varepsilon$ ). За приближенное значение можно принять середину отрезка  $[a, b]$ .

При  $n$  итерациях получим соотношение  $(b - a)/2^n \leq \varepsilon$  из которого можно вычислить число итераций, необходимое для достижения заданной степени точности  $n \geq \ln_2(b - a)/\varepsilon$ .

Геометрическая интерпретация.

Решим нелинейное уравнение:  $0,186 \cdot \cos \sigma - 4,18 \cdot \sigma^2 + 0,124 = 0$ , заменив переменную  $\sigma$  на  $x$ . Т.е.  $f(x) = 0,186 \cdot \cos x - 4,18 \cdot x^2 + 0,124$

Для локализации корня строим таблицу и график. Очевидно, что корень находится на отрезке  $[0,2; 0,3]$



Программа (макрос):

```
Sub bis()
```

```
Dim a As Single, b As Single, eps As Single
```

```
a = Range("a").Value
```

```
b = Range("b").Value
```

```
eps = Range("eps").Value
```

```
10 fa = fnf(a)
```

```
fb = fnf(b)
```

```
c = (a + b) / 2!
```

```
fc = fnf(c)
```

```
If fc = 0 Then GoTo 20
```

```
If Abs(a - b) < eps Then GoTo 20
```

```
If fa * fc < 0 Then b = c Else a = c
```

```
GoTo 10
```

```
20 Range("c6").Value = c
```

```
Range("fc").Value = fc
```

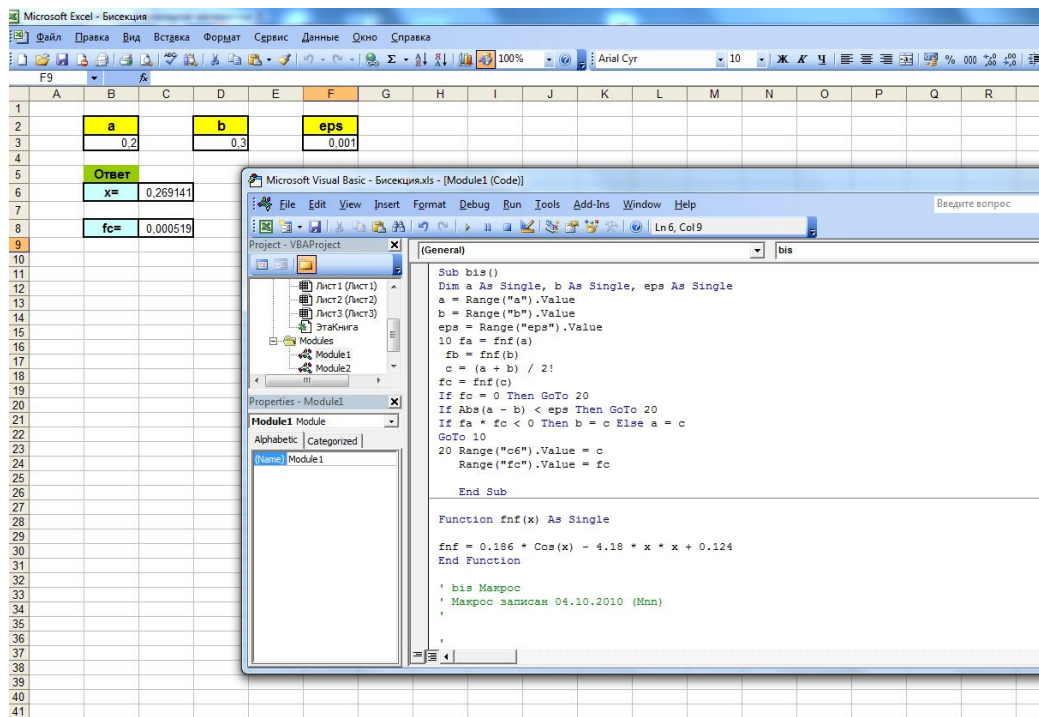
```
End Sub
```

```
Function fnf(x) As Single
```

```
fnf = 0.186 * Cos(x) - 4.18 * x * x + 0.124
```

```
End Function
```





Результат расчета:  $x = 0.269$

Т.е. значение коэффициента  $\sigma = 0,269$ .

## 2. Метод простой итерации (последовательных приближений).

Этот метод является наиболее общим и многие другие методы можно представить как некоторую вариацию метода простых итераций.

Для применения этого метода уравнение  $f(x) = 0$  (1) преобразуется к виду:

$$x = \varphi(x). \quad (3)$$

Это можно сделать, различными способами, например,

$x = x + c \cdot f(x)$ ,  $c \neq 0$ .  $\varphi(x)$  – итерационная функция.

Предположим, что выбрано некоторое начальное приближение корня уравнения (1)  $x_0$  ( $x_0 \in [a, b]$ ). Подставим его в правую часть уравнения (3). Тогда получим некоторое число

$$x_1 = \varphi(x_0)$$

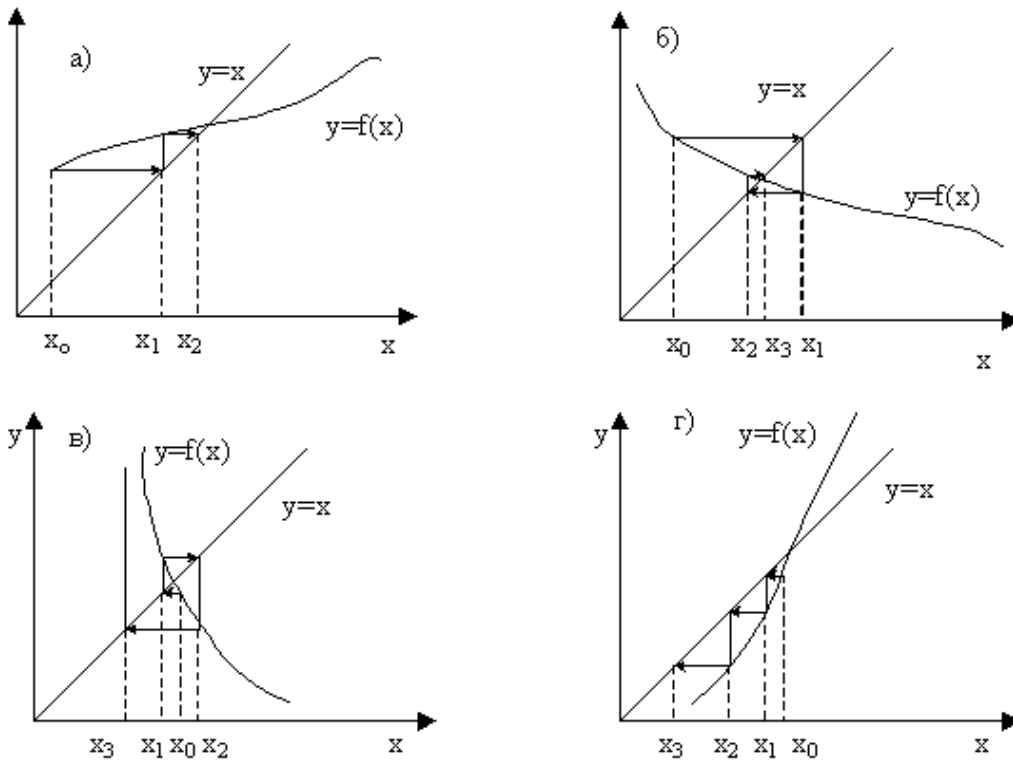
Подставляя теперь в правую часть равенства (1) вместо  $x_0$  число  $x_1$ , получим новое число  $x_2 = \varphi(x_1)$ . Повторяя этот процесс, будем иметь последовательность чисел

$$x_n = \varphi(x_{n-1}), \quad n = 0, 1, 2, \dots \quad (4)$$

Такая последовательность называется итерационной.

Если эта последовательность - сходящаяся, т. е. существует предел  $x^* = \lim_{n \rightarrow \infty} x_n$ , то этот предел является корнем уравнения и может быть вычислен по формуле (3) с любой степенью точности.

Графическая интерпретация метода: решением уравнения (3) будет абсцисса точки пересечения прямой  $y=x$  с кривой  $y = \varphi(x)$ .



При применении данного метода встает вопрос об условиях его сходимости.

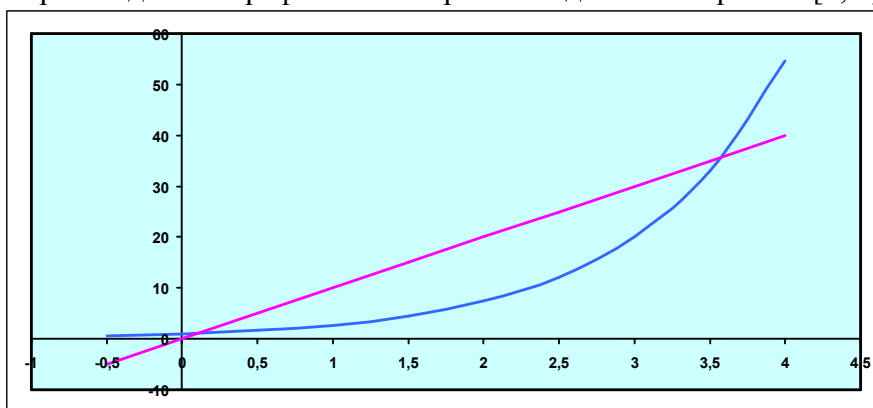
**Теорема.** Пусть функция  $\varphi(x)$  определена и непрерывно дифференцируема на некотором отрезке  $[a,b]$  и ее производная  $|\varphi'(x)| \leq q$ , где  $0 \leq q < 1$ , тогда итерационная последовательность сходится к единственному на  $[a,b]$  корню уравнения.

Скорость сходимости метода зависит от величины  $q$ . Чем она меньше, тем быстрее сходится метод. Поэтому на практике при нахождении корней методом простой итерации стараются таким образом подобрать параметр  $c$  в формуле, чтобы производная  $\varphi'(x)$  в окрестности корня по абсолютной величине была как можно меньше.

Критерием окончания расчета при заданной точности  $\varepsilon$  можно считать выполнение неравенства  $|x_n - x_{n-1}| < \varepsilon$ .

Пример 1. Найти корни уравнения  $e^x - 10x = 0$  с точностью  $\varepsilon = 0,0001$ .

Корни отделяем графически. Корни находятся на отрезках  $[0, 1]$  и  $[3, 4]$ .



Преобразуем уравнение к виду:  $x = 1/10 \cdot e^x$

Условия сходимости выполняются на отрезке  $[0, 1]$ . За начальное приближение примем  $x_0 = 1$ .

Запишем результаты итераций в таблицу.

N итерации	$x_n$	$ x_n - x_{n-1} $
0	1	
1	0,271828	0,72817
2	0,131236	0,14059
3	0,114024	0,01721
4	0,112078	0,00195
5	0,11186	0,00022
6	0,111836	2,4E-05

Корень уравнения  $x = 0,11183$ .

Пример 2. Решим уравнение  $e^x \cdot \sin x - 1 = 0$ . Отрезок локализации корня  $[0; \pi/2]$ .

Для того чтобы подобрать итерационную функцию будем представлять уравнение в виде  $x = x + c \cdot f(x)$ .

$$x = x + c(e^x \cdot \sin x - 1)$$

$$\varphi(x) = x + c(e^x \cdot \sin x - 1)$$

Для того чтобы условие сходимости выполнялось, необходимо таким образом подобрать параметр  $c$ , чтобы  $|\varphi'(x)| \leq 1$ .

$$|\varphi'(x)| = |1 + ce^x(\sin x + \cos x)| < 1$$

Очевидно, что условие не может выполняться, если  $c > 0$ .

Чтобы понять как ведет себя производная функции  $\varphi(x)$ , найдем  $\varphi''(x)$ .

$$\varphi''(x) = 2ce^x \cos x$$

На интервале  $[0; \pi/2]$   $e^x > 0$ ,  $\cos x > 0$ ,  $c < 0$ . Т.е.  $\varphi''(x) < 0$ . Следовательно  $\varphi'(x)$  убывает. Тогда наибольшее значение функции  $\varphi'(x)$  на интервале при  $x = 0$ :  $\varphi'(0) = 1 + c$ .

А наименьшее значение при  $x = \frac{\pi}{2}$ :  $\varphi'\left(\frac{\pi}{2}\right) \approx 1 + 4,8c$ .

$1 + c = 0 \Rightarrow c = -1$ ,  $1 + 4,8c = 0 \Rightarrow c \approx -0,21$ . В качестве  $c$  выберем среднее арифметическое двух значений:  $c = -0,6$ .

$$\varphi(x) = x - 0,6(e^x \cdot \sin x - 1)$$

За начальное приближение выберем середину отрезка локализации  $x_0 = 0,7$ .

N итерации	$x_n$	$ x_n - x_{n-1} $
0	0,7	
1	0,521623	0,178377
2	0,617926	0,096303
3	0,573088	0,044838
4	0,596025	0,022938
5	0,584744	0,011282
6	0,59041	0,005667
7	0,587593	0,002818
8	0,589001	0,001408
9	0,588299	0,000702
10	0,588649	0,00035
11	0,588475	0,000175
12	0,588562	8,73E-05

Корень уравнения  $x = 0,5886$ .

### 3. Метод Ньютона (касательных).

Зададим некоторое начальное приближение  $x_0 \in [a, b]$  и линеаризуем функцию  $f(x)$  в окрестности  $x_0$  с помощью отрезка ряда Тейлора

$$f(x) = f(x_0) + f'(x_0)(x - x_0). \quad (5)$$

Вместо уравнения (1) решим линеаризованное уравнение

$$f(x_0) + f'(x_0)(x - x_0) = 0,$$

трактуя его решение  $x$  как первое приближение к корню

$$x_1 = x_0 - f(x_0)/f'(x_0).$$

Продолжая этот процесс, приходим к формуле Ньютона

$$x_{n+1} = x_n - f(x_n)/f'(x_n) \quad (6),$$

которую можно считать итерационным процессом с итерирующей функцией  $\varphi(x) = x - f(x)/f'(x)$ .

Геометрическая интерпретация этого процесса показана на рис. Уравнение (5) является уравнением линии касательной к кривой  $f(x)$  в точке  $x_0$ , поэтому этот метод называют методом касательных. Формулу (6) можно легко вывести также из геометрических соображений.

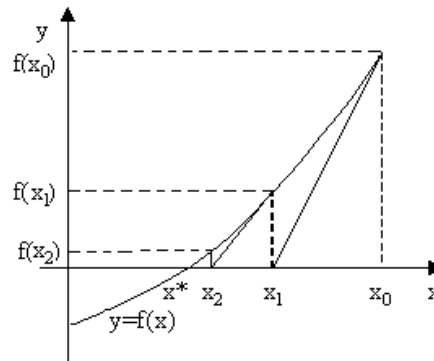


Рис.

Если нулевое приближение выбрано достаточно близко к корню, ньютоновские итерации **сходятся**. Скорость сходимости квадратичная, то есть на каждом шаге погрешность пропорциональна квадрату предыдущей.

Следует отметить, что улучшение сходимости этого метода по сравнению с предыдущими достигается увеличением затрат на выполнение каждого шага, так как в каждом шаге требуется вычислять не только значение функции  $f(x)$ , но и ее производной  $f'(x)$ .

Практический критерий окончания расчета при заданной точности  $\varepsilon$  - выполнение неравенства  $|x_n - x_{n-1}| < \varepsilon$ .

Для нахождения корней произвольной дифференцируемой функции чаще всего применяют метод Ньютона, особенно если известны разумные начальные приближения для корней.

**Пример.** Рассмотрим решение уравнения  $f(x) \equiv x^2 - a = 0$ .

Формула метода Ньютона принимает вид:

$$x_{n+1} = x_n - \frac{x_n^2 - a}{2x_n} = \frac{1}{2} \left( x_n + \frac{a}{x_n} \right)$$

Мы получили формулу, которая позволяет очень быстро находить квадратный корень только с помощью умножения и деления.

Вот, например  $\sqrt{5}$ , найденный по этой формуле.

N итерации	$x_n$
0	1
1	3
2	2,333333
3	2,238095
4	2,236069
5	2,236068
6	2,236068
7	2,236068

### Решение систем линейных уравнений.

Пусть дана система  $n$  линейных алгебраических уравнений с  $n$  неизвестными.

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n = b_1 \\ a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n = b_2 \\ \dots \\ a_{n1}x_1 + a_{n2}x_2 + \dots + a_{nn}x_n = b_n \end{cases} \quad (1)$$

и пусть  $\det A \neq 0$  (определитель матрицы системы  $\neq 0$ ). В этом случае система имеет единственное решение.

Тогда в матричной форме  $A \cdot X = B$ .

На практике кроме существования и единственности решения важна еще устойчивость относительно погрешностей правой части и элементов матрицы. Иногда небольшое изменение исходных данных может сильно изменить решение. В этом случае систему называют плохо обусловленной. Очевидно, у плохо обусловленных систем  $\det A \approx 0$ . Геометрически плохо обусловленная система из 2-х уравнений соответствует почти параллельным прямым.

#### Пример.

Рассмотрим систему уравнений:  $\begin{cases} 5x + 7y = 12 \\ 7x + 10y = 17 \end{cases}$ . Эта система имеет единственное

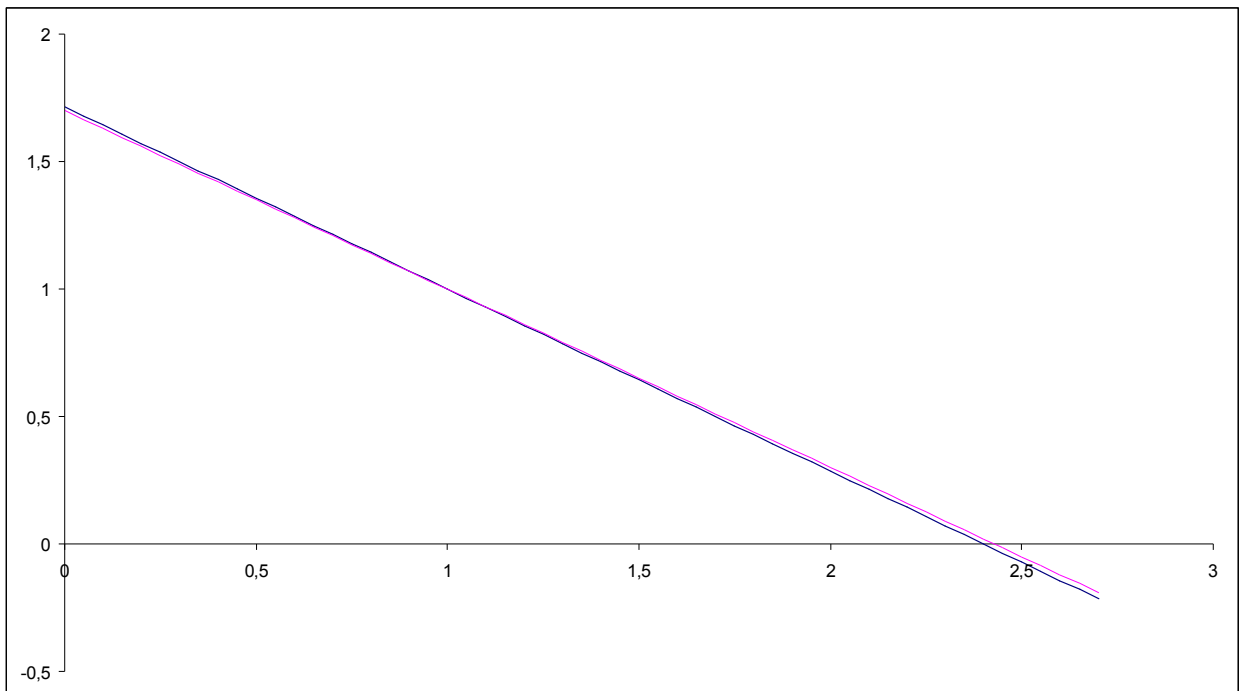
решение  $x = 1$  и  $y = 1$ .

Теперь возьмем значения неизвестных  $x = 2,415$  и  $y = 0$ . Подставим эти значения в исходную систему.

$$\begin{cases} 5x + 7y = 12,075 \\ 7x + 10y = 16,905 \end{cases}$$

Если округлить правые части этой системы до двух знаков, то она будет совпадать с исходной. И если исходные данные будут заданы с такой точностью, то это решение так же хорошо отвечает условиям поставленной задачи как и  $x = 1$  и  $y = 1$ .

На графике видно, что прямые заданные уравнениями системы практически параллельны.





© Лекции подготовлены доц. Мусиной М.В.

$$a_{ij}^{(k)} = a_{ij}^{(k-1)} + m_{ik} a_{kj}, \quad b_i^{(k)} = b_i^{(k-1)} + m_{ik} b_{k-1}^{(k-1)}, \quad \text{где } m_{ik} = -a_{ik}^{(k-1)} / a_{kk}^{(k-1)} \quad (i, j = k+1, \dots, n)$$

После  $n-1$  шага вычисления прямого хода заканчиваются и, последовательно исключив все неизвестные, получим систему треугольного вида:

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \dots + a_{1k}x_k + \dots + a_{1n}x_n &= b_1, \\ a_{22}^{(1)}x_2 + \dots + a_{2k}^{(1)}x_k + \dots + a_{2n}^{(1)}x_n &= b_2^{(1)} \\ &\dots\dots\dots \\ a_{n-1,n-1}^{(n-1)}x_{n-1} + a_{n-1,n}^{(n-1)}x_n &= b_{n-1}^{(n-1)} \\ a_{nn}^{(n-1)}x_n &= b_n^{(n-1)} \end{aligned}$$

Заметим, что выполнение прямого хода было возможно при условии, что все  $a_{ii}^{(l)}$ ,  $l=1,2,\dots,n-1$  не равны нулю. Если условие не выполняется, то необходимо переставить строки так, чтобы оно выполнялось.

**Обратный ход.**

Из последнего уравнения системы легко найти  $x_n$ . 
$$x_n = \frac{b_n^{(n-1)}}{a_{nn}^{(n-1)}}$$

Выполняя последовательные подстановки в системе можно получить все значения неизвестных  $x_{n-1}, x_{n-2}, \dots, x_1$  по формулам:

$$x_i = \frac{1}{a_{ii}^{(i-1)}} (b_i^{(i-1)} - \sum_{j=i+1}^n a_{ij}^{(i-1)} x_j) \quad (i = n-1, n-2, \dots, 1)$$

Эта процедура получила название обратный ход метода Гаусса.

**Пример.**

$$\begin{cases} x_1 + 0,17x_2 + 0,25x_3 + 0,54x_4 = 0,3 \\ 0,47x_1 + x_2 + 0,67x_3 - 0,32x_4 = 0,5 \\ -0,11x_1 + 0,35x_2 + x_3 - 0,74x_4 = 0,7 \\ -0,55x_1 + 0,43x_2 + 0,36x_3 + x_4 = 0,9 \end{cases}$$

Решение:

Запишем в таблицу расширенную матрицу системы, с которой будем производить действия.

1	0,17	0,25	0,54	0,3
0,47	1	0,67	-0,32	0,5
-0,11	0,35	1	-0,74	0,7
-0,55	0,43	0,36	1	0,9

Последовательно умножим первую строку на 0,47, -0,11 и -0,55 и отнимем соответственно из 2, 3 и 4 строки.

Преобразованная матрица:

1	0,17	0,25	0,54	0,3
0	0,9201	0,5525	-0,5738	0,359
0	0,3687	1,0275	-0,6806	0,733
0	0,5235	0,4975	1,297	1,065

Вторую строку разделим на 0,92 и получим множители второй строки

0	1	0,600478	-0,62363	0,390175
---	---	----------	----------	----------

Далее последовательно умножаем эту строку на 0,36 и 0,52 и вычитаем из 3-й и 4-й строк.

1	0,17	0,25	0,54	0,3
0	1	0,600478	-0,62363	0,390175
0	0	0,806104	-0,45067	0,589142
0	0	0,18315	1,623469	0,860743

Множители третьей строки:

0	0	1	-0,55907	0,730852
---	---	---	----------	----------

Эту строку умножаем на 0,183 и вычитаем из 4-й строки.

1	0,17	0,25	0,54	0,3
---	------	------	------	-----

0	1	0,600478	-0,62363	0,390175
0	0	1	-0,55907	0,730852
0	0	0	1,725863	0,726888

Теперь разделим элементы 4-й строки на 1,72 и получаем матрицу треугольного вида:

1	0,17	0,25	0,54	0,3
0	1	0,600478	-0,62363	0,390175
0	0	1	-0,55907	0,730852
0	0	0	1	0,421174

Осуществляем обратный ход метода Гаусса.

$$x_1 = 0,181; x_2 = 0,0725; x_3 = 0,966; x_4 = 0,421$$

Метод Гаусса может быть легко реализован на компьютере. При выполнении вычислений, как правило, не интересуют промежуточные значения матрицы  $A$ . Поэтому численная реализация метода сводится к преобразованию элементов массива размерности  $(n \times (n+1))$ , где  $n+1$  столбец содержит элементы правой части системы.

Один из основных недостатков метода Гаусса связан с тем, что при его реализации накапливается вычислительная погрешность. Для того, чтобы уменьшить рост вычислительной погрешности применяются различные модификации метода Гаусса. Например, **метод Гаусса с выбором главного элемента по столбцам**, в этом случае на каждом этапе прямого хода строки матрицы переставляются таким образом, чтобы диагональный угловой элемент был максимальным. При исключении соответствующего неизвестного из других строк деление будет производиться на наибольший из возможных коэффициентов и следовательно относительная погрешность будет наименьшей.

Существует **метод Гаусса с выбором главного элемента по всей матрице**. В этом случае переставляются не только строки, но и столбцы. Использование модификаций метода Гаусса приводит к усложнению алгоритма увеличению числа операций и соответственно к росту времени счета.

Модифицированные методы Гаусса предлагается изучить самостоятельно, как дополнительный материал.

Выполняемые в методе Гаусса преобразования прямого хода, приведшие матрицу  $A$  системы к треугольному виду позволяют вычислить определитель матрицы

$$\det A = \begin{vmatrix} a_{11} & a_{12} & \dots & a_{1m} \\ 0 & a_{22}^{(1)} & \dots & a_{2m}^{(1)} \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & a_{m,m}^{(m-1)} \end{vmatrix} = a_{11} \cdot a_{22}^{(1)} \dots a_{m,m}^{(m-1)}.$$

Метод Гаусса также позволяет найти обратную матрицу.

### Решение систем линейных уравнений методом простых итераций.

Если система имеет большую размерность ( $10^3 - 10^6$  уравнений) или матрица системы **разрежена**, более эффективны для решения не прямые итерационные методы. Разреженная матрица возникает в системе, где многие коэффициенты при неизвестных равны нулю.

Итерационные методы (методы последовательных приближений) состоят в том, что решение системы (1) находится как предел последовательных приближений  $\bar{x}^{(n)}$  при  $n \rightarrow \infty$ , где  $n$  номер итерации. При использовании методов итерации обычно задается некоторое малое число  $\varepsilon > 0$  и вычисления проводятся до тех пор, пока не будет выполнена оценка  $\|\bar{x}^{(n)} - \bar{x}\| < \varepsilon$ . К этим методам относятся метод Зейделя, Якоби, метод верхних релаксаций и т.д.

Рассмотрим один из простых методов – метод простой итерации.



Пусть дана система линейных уравнений, имеющая вид:

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n = b_1 \\ a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n = b_2 \\ \dots \\ a_{n1}x_1 + a_{n2}x_2 + \dots + a_{nn}x_n = b_n \end{cases} \quad (1).$$

Эта система эквивалентна векторной (матричной) записи  $A \cdot \vec{X} = \vec{B}$ , где  $A$  – матрица системы,

$$\vec{X} = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} - \text{вектор – столбец неизвестных,}$$

$$\vec{B} = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{pmatrix} - \text{вектор – столбец свободных членов.}$$

Данную систему можно преобразовать к виду  $\vec{X} = C \cdot \vec{X} + \vec{F}$  (2), где  $C$  – некоторая матрица,  $\vec{F}$  – вектор – столбец.

Обозначим  $C \cdot \vec{X} + \vec{F} = \Phi(\vec{X})$ , то есть система принимает форму  $\vec{X} = \Phi(\vec{X})$ . То есть, начиная с некоторого начального значения (нулевого приближения) получим последовательность.

$$\vec{X}^{(1)} = \Phi(\vec{X}^{(0)})$$

$$\vec{X}^{(2)} = \Phi(\vec{X}^{(1)})$$

...

$$\vec{X}^{(k+1)} = \Phi(\vec{X}^{(k)}).$$

Аналогичную последовательность мы получали при решении нелинейного уравнения методом простой итерации  $x = \varphi(x)$ .

Возникает только вопрос о том, как привести систему (1) к виду *удобному для итераций* (2). В общем случае это не простая задача, требующая специальных знаний, но в некоторых случаях можно поступить очень просто. Например, можно сделать такие преобразования:

$$A \cdot \vec{X} = (A - I + I) \cdot \vec{X} = \vec{B}, \text{ где } I - \text{единичная матрица.}$$

$$(A - I) \cdot \vec{X} + I \cdot \vec{X} = \vec{B}$$

$$\vec{X} = (I - A) \cdot \vec{X} + \vec{B}$$

Самый простой способ следующий. Из первого уравнения системы (1) выразим неизвестное  $x_1$ :

$$x_1 = \frac{1}{a_{11}}(b_1 - a_{12}x_2 - a_{13}x_3 \dots - a_{1n}x_n)$$

Из второго уравнения  $x_2$ :

$$x_2 = \frac{1}{a_{22}}(b_2 - a_{21}x_1 - a_{23}x_3 \dots - a_{2n}x_n) \text{ и так далее. В результате получим систему:}$$

$$\begin{cases} x_1 = c_{12}x_2 + c_{13}x_3 + \dots + c_{1n}x_n + f_1 \\ x_2 = c_{21}x_1 + c_{23}x_3 + \dots + c_{2n}x_n + f_2 \\ \dots \\ x_n = c_{n1}x_1 + c_{n2}x_2 + c_{n3}x_3 + \dots + \dots + f_n \end{cases}$$

На главной диагонали матрицы  $C$  стоят нули, а ненулевые элементы выражают по формулам:

$$c_{ij} = -\frac{a_{ij}}{a_{ii}}, \quad f_i = \frac{b_i}{a_{ii}} \quad (i, j = 1, 2, \dots, n; i \neq j) \quad (3)$$

Для выполнения данного преобразования необходимо чтобы диагональные элементы исходной матрицы не были нулевыми. Если матрица преобразована к виду (2) по данным формулам (3), то метод итераций называют **методом Якоби**.

Для системы (1) метод итерации сходится, если модули диагональных коэффициентов для каждого уравнения системы больше суммы модулей всех остальных коэффициентов (не считая свободных членов).

В качестве критерия окончания расчета (достижения заданной точности) можно использовать простое соотношение  $\left| \vec{X}^{(n+1)} - \vec{X}^{(n)} \right| < \varepsilon$ .

**Пример.** Решим систему уравнений с точностью  $\varepsilon = 0,001$

$$\begin{cases} 6,25x_1 - x_2 + 0,5x_3 = 7,5 \\ -x_1 + 5x_2 + 2,12x_3 = -8,68 \\ 0,5x_1 + 2,12x_2 + 3,6x_3 = -0,24 \end{cases}$$

Перепишем систему пересчитав коэффициенты по формулам (3).

$$\begin{cases} x_1 = 0,16x_2 - 0,08x_3 + 1,2 \\ x_2 = 0,2x_1 - 0,424x_3 - 1,786 \\ x_3 = -0,1389x_1 - 0,58889x_2 - 0,0667 \end{cases}$$

$$\text{Здесь } C = \begin{pmatrix} 0 & 0,16 & -0,08 \\ 0,2 & 0 & -0,424 \\ -0,1389 & -0,5889 & 0 \end{pmatrix}, \quad F = \begin{pmatrix} 1,2 \\ -1,786 \\ -0,0667 \end{pmatrix}.$$

$$\text{За начальное приближение примем } \vec{X}^{(0)} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}.$$

Результаты итераций запишем в таблицу.

$n$	0	1	2	3	4
$x_1^{(n)}$	0	1,2	0,9276	0,902	0,8449
$x_2^{(n)}$	0	-1,736	-1,4677	-1,885	-1,8392
$x_3^{(n)}$	0	-0,0667	0,789	0,6688	0,9181
$\left  \vec{X}^{(n+1)} - \vec{X}^{(n)} \right $		1,736	0,8557	0,4173	0,2493

$n$	...	12	13	14	15
$x_1^{(n)}$	...	0,8006	0,8003	0,8002	0,8001
$x_2^{(n)}$	...	-1,9985	-1,9993	-1,9995	-1,9998
$x_3^{(n)}$	...	0,9987	0,9990	0,9995	0,9997

$ \vec{X}^{(n+1)} - \vec{X}^{(n)} $	...	0,0018	0,0008	0,0005	0,0003
-------------------------------------	-----	--------	--------	--------	--------

После 15 итераций достигнута искомая точность, и можно записать решение.  $x_1 = 0,8$ ;  $x_2 = -2,0$ ;  $x_3 = 1,0$ .

### Методы численной интерполяции и аппроксимации.

Процесс научного познания часто приводит к необходимости обработки числовой информации. При обращении с данными научного и инженерного характера особенно важными являются такие вычислительные средства как **интерполяция и приближение кривыми (аппроксимация)**.

Некоторые типичные ситуации:

1. Функция  $f$  задана таблицей своих значений:  $y_i = f(x_i)$ . Требуется найти значение в точке  $x \neq x_i$ .
2. Функция  $f$  имеет громоздкое выражение и требует для вычисления значительных затрат времени.
3. Значения функции  $f$  найдены экспериментально с погрешностью, т.е.  $y_i^*$  вместо  $y_i$ .

Возникающие проблемы решают следующим образом: функцию  $f(x)$  заменяют на  $g(x)$ , значения которой принимают за приближенные значения функции  $f(x)$ .

#### Линейная интерполяция.

Простейшим видом интерполяции является **кусочно-линейная интерполяция**. Она состоит в том, что заданные точки  $(x_i, y_i)$ ,  $i = 0, \dots, n$  соединяются прямолинейными отрезками, и функция  $f(x)$  приближается полученной ломаной.

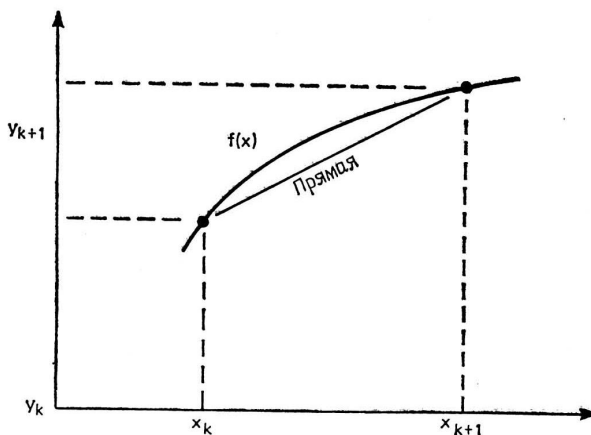
Уравнения каждого отрезка ломаной в общем случае разные. Для каждого из интервалов  $(x_{i-1}, x_i)$  в качестве уравнения интерполянты используется уравнения прямой, проходящей через две точки. В частности, для  $i$ -го интервала можно написать уравнение прямой, проходящей через точки  $(x_{i-1}, y_{i-1})$  и  $(x_i, y_i)$ , в виде

$$(y - y_{i-1}) / (y_i - y_{i-1}) = (x - x_{i-1}) / (x_i - x_{i-1})$$

Отсюда

$$y = a_i x + b_i \equiv g(x), x \in (x_{i-1}, x_i)$$

$$a_i = (y_i - y_{i-1}) / (x_i - x_{i-1}), b_i = y_{i-1} - a_i x_{i-1}, i = 1, \dots, n.$$



Следовательно, при использовании кусочно-линейной интерполяции сначала нужно определить номер  $i$  интервала, в который попадает значение аргумента  $x$ , затем подставить  $x$  и  $i$  в формулу и найти приближенное значение функции в точке  $x$ .

Обычно полагают, что аппроксимируя истинную кривую более сложной линией можно уточнить полученный результат. Например, можно найти многочлен  $n$ -ной степени  $P_n(x)$ ,

аппроксимирующий функцию  $f(x)$  кривой, проходящей через все  $n + 1$ , заданные в таблице точки  $(x_i, y_i)$ , где  $i = 0, \dots, n$ . В этом случае многочлен должен удовлетворять условиям  $P_n(x_i) = y_i$ , для всех  $i$ .

Точки  $(x_i, y_i)$  называют **узлами интерполирования**, а искомый многочлен – **интерполяционным**.

### Интерполяция многочленами.

Одна из форм записи интерполяционного многочлена - **многочлен Лагранжа**.

Интерполяционный многочлен для этого метода запишем в виде.

$P_n(x) = y_0 b_0(x) + y_1 b_1(x) + \dots + y_n b_n(x)$ , где все  $b_j(x)$  – многочлены степени  $n$ , коэффициенты которых можно найти с помощью  $n + 1$  уравнений  $P_n(x_i) = y_i, i = 0, \dots, n$ .

Подставим данные условия в интерполяционный многочлен и получим систему уравнений:

$$y_0 b_0(x_0) + y_1 b_1(x_0) + \dots + y_n b_n(x_0) = y_0$$

...

$$y_0 b_0(x_n) + y_1 b_1(x_n) + \dots + y_n b_n(x_n) = y_n$$

Если значения  $b_j(x_i)$  выбраны так, что

$$b_j(x_i) = \begin{cases} 1 & \text{при } i = j \\ 0 & \text{при } i \neq j \end{cases}, \text{ то уравнения будут удовлетворены.}$$

Это означает, что любой многочлен  $b_j(x)$  равен нулю во всех точках  $x_i$  кроме  $x_j$  и его можно представить в виде:

$$b_j(x) = C_j (x - x_0)(x - x_1) \dots (x - x_{j-1})(x - x_{j+1}) \dots (x - x_n)$$

Так как  $b_j(x_j) = 1$ , то коэффициент  $C_j$  определяется выражением

$$C_j = 1 / ((x_j - x_0)(x_j - x_1) \dots (x_j - x_{j-1})(x_j - x_{j+1}) \dots (x_j - x_n))$$

Т.е. многочлен

$$b_j(x) = \frac{(x - x_0)(x - x_1) \dots (x - x_{j-1})(x - x_{j+1}) \dots (x - x_n)}{(x_j - x_0)(x_j - x_1) \dots (x_j - x_{j-1})(x_j - x_{j+1}) \dots (x_j - x_n)}$$
 является требуемым мно-

гчленом степени. Подставим все в формулу для  $P_n(x)$  и получим.

$$P_n(x) = \sum_{j=0}^n \frac{(x - x_0)(x - x_1) \dots (x - x_{j-1})(x - x_{j+1}) \dots (x - x_n)}{(x_j - x_0)(x_j - x_1) \dots (x_j - x_{j-1})(x_j - x_{j+1}) \dots (x_j - x_n)} y_j = L_n(x)$$

Это интерполяционный многочлен Лагранжа.

В инженерной практике наиболее часто используется интерполяция многочленами первой, второй и третьей степени (*линейная, квадратичная и кубическая интерполяции*).

При  $n = 1$  – линейная формула.

$$L_1(x) = y_0 \frac{x - x_1}{x_0 - x_1} + y_1 \frac{x - x_0}{x_1 - x_0}. \text{ (Совпадает с уже полученной формулой линейной ин-}$$

терполяции по двум узлам  $x_0, x_1$ .)

При  $n = 2$

$$L_2(x) = y_0 \frac{(x - x_1)(x - x_2)}{(x_0 - x_1)(x_0 - x_2)} + y_1 \frac{(x - x_0)(x - x_2)}{(x_1 - x_0)(x_1 - x_2)} + y_2 \frac{(x - x_0)(x - x_1)}{(x_2 - x_0)(x_2 - x_1)}.$$

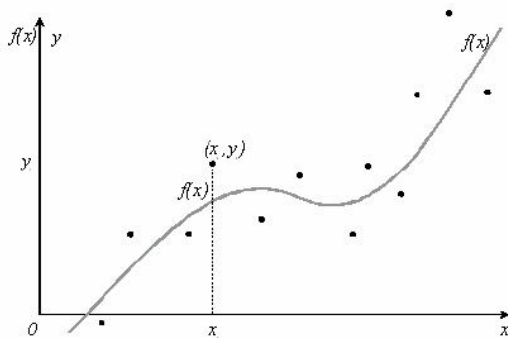
Существует и другой подход для построения интерполяционного многочлена — метод Ньютона (метод разделённых разностей). Но фактически формулы Лагранжа и Ньютона порождают один и тот же полином, разница только в алгоритме его построения.

### Аппроксимация функций по методу наименьших квадратов.

В инженерной деятельности часто возникает необходимость описать в виде функциональной зависимости связь между величинами, заданными таблично или в виде набора точек  $(x_i, y_i)$ , где  $i = 0, \dots, n$ . Как правило, эти табличные данные получены экспериментально и имеют погрешности.

При аппроксимации желательно получить относительно простую функциональную зависимость (например, многочлен), которая позволила бы «сгладить» экспериментальные погрешности, вычислять значения функции в точках не содержащихся в исходной таблице.

Эта функциональная зависимость должна с достаточной точностью соответствовать исходной табличной зависимости. В качестве критерия точности чаще всего используют критерий *наименьших квадратов*, т.е. определяют такую функциональную зависимость  $f(x)$ , при которой  $R = \sum_{i=0}^n (y_i - f(x_i))^2$  обращается в минимум.



Рассмотрим в качестве функциональной зависимости многочлен.

$$P_m(x) = a_0 + a_1x + a_2x^2 + \dots + a_mx^m, \text{ тогда } R = \sum_{i=0}^n (y_i - P_m(x_i))^2.$$

Условия минимума – нулевые частные производные по всем переменным  $a_0, a_1, a_2, \dots$

$$a_m. \text{ Т.е. } \frac{\partial R}{\partial a_k} = -\sum_{i=0}^n 2(y_i - a_0 - a_1x_i - \dots - a_mx_i) x_i^k = 0, \text{ или}$$

$$\sum_{i=0}^n (y_i - a_0 - a_1x_i - \dots - a_mx_i) x_i^k = 0, \quad k = 0, 1, 2, \dots, m.$$

Собираем коэффициенты при неизвестных  $a_0, a_1, a_2, \dots, a_m$ , получаем систему уравнений:

$$a_0 \sum_{i=0}^n x_i^k + a_1 \sum_{i=0}^n x_i^{k+1} + a_2 \sum_{i=0}^n x_i^{k+2} + \dots + a_m \sum_{i=0}^n x_i^{k+m} = \sum_{i=0}^n x_i^k y_i, \quad k = 0, 1, 2, \dots, m.$$

Можно ввести обозначения:

$$c_k = \sum_{i=0}^n x_i^k, \quad b_k = \sum_{i=0}^n x_i^k y_i \text{ и переписать систему в развернутом виде.}$$

$$\begin{cases} c_0 a_0 + c_1 a_1 + c_2 a_2 + \dots + c_m a_m = b_0 \\ c_1 a_0 + c_2 a_1 + c_3 a_2 + \dots + c_{m+1} a_m = b_1 \\ \dots \\ c_m a_0 + c_{m+1} a_1 + c_{m+2} a_2 + \dots + c_{2m} a_m = b_m \end{cases}$$

Матрица данной системы называется матрицей Грамма. Решая эту систему линейных уравнений, получаем коэффициенты  $a_0, a_1, a_2, \dots, a_m$ , которые являются искомыми параметрами эмпирической формулы.

Рассмотрим два частных случая  $m = 1$  и  $m = 2$ .

1. Линейная аппроксимация ( $m = 1$ ).

$$P_1(x) = a_0 + a_1x$$

$$c_0 = \sum_{i=0}^n x_i^0 = n + 1, \quad c_1 = \sum_{i=0}^n x_i, \quad c_2 = \sum_{i=0}^n x_i^2, \quad b_0 = \sum_{i=0}^n y_i, \quad b_1 = \sum_{i=0}^n x_i y_i$$

Таким образом наша система уравнений имеет вид:

$$\begin{cases} (n+1)a_0 + \sum_{i=0}^n x_i a_1 = \sum_{i=0}^n y_i \\ \sum_{i=0}^n x_i a_0 + \sum_{i=0}^n x_i^2 a_1 = \sum_{i=0}^n x_i y_i \end{cases}$$

Решаем ее методом Крамера.

$$\Delta = \begin{vmatrix} n+1 & \sum_{i=0}^n x_i \\ \sum_{i=0}^n x_i & \sum_{i=0}^n x_i^2 \end{vmatrix}, \quad a_0 = \frac{\begin{vmatrix} \sum_{i=0}^n y_i & \sum_{i=0}^n x_i \\ \sum_{i=0}^n x_i y_i & \sum_{i=0}^n x_i^2 \end{vmatrix}}{\Delta}, \quad a_1 = \frac{\begin{vmatrix} n+1 & \sum_{i=0}^n y_i \\ \sum_{i=0}^n x_i & \sum_{i=0}^n x_i y_i \end{vmatrix}}{\Delta}$$

И получаем искомую функцию  $y = a_0 + a_1 x$ .

2. Квадратичная аппроксимация ( $m = 2$ ).

$$P_2(x) = a_0 + a_1 x + a_2 x^2$$

Кроме  $c_0, c_1, c_2, b_0, b_1$  рассчитываются

$$c_3 = \sum_{i=0}^n x_i^3, \quad c_4 = \sum_{i=0}^n x_i^4, \quad b_2 = \sum_{i=0}^n y_i x_i^2.$$

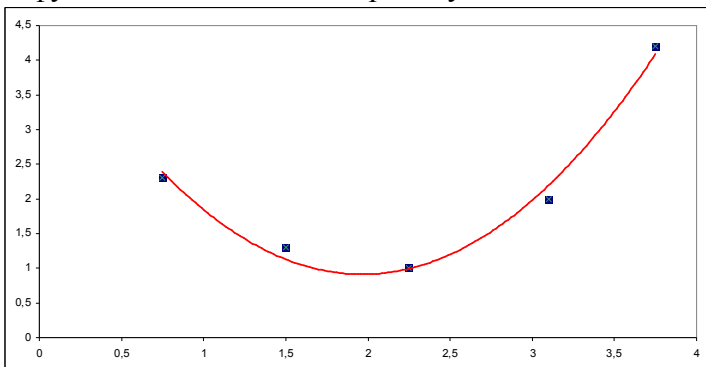
Расширенная матрица системы уравнений:  $\left( \begin{array}{ccc|c} c_0 & c_1 & c_2 & b_0 \\ c_1 & c_2 & c_3 & b_1 \\ c_2 & c_3 & c_4 & b_2 \end{array} \right)$ , решив которую полу-

чим искомые коэффициенты  $a_0, a_1, a_2$ .

**Пример.** Получить эмпирическую формулу для функции  $f(x)$ , заданной таблицей, используя метод наименьших квадратов.

$x$	0,75	1,5	2,25	3	3,75
$f(x)$	2,3	1,3	1,0	2,2	4,2

Табличные данные изобразим на графике, из которого видно, что в качестве эмпирической функции можно взять параболу.



Найдем коэффициенты системы уравнений из таблицы.

$n$	$x$	$y$	$x^2$	$x^3$	$x^4$	$xy$	$x^2y$
0	0,75	2,3	0,5625	0,421875	0,316406	1,725	1,29375
1	1,5	1,3	2,25	3,375	5,0625	1,95	2,925
2	2,25	1	5,0625	11,39063	25,62891	2,25	5,0625
3	3	2,2	9	27	81	6,6	19,8
4	3,75	4,2	14,0625	52,73438	197,7539	15,75	59,0625
$\sum_{i=0}^n$	11,25	11	30,9375	94,92188	309,7617	28,275	88,14375

$$\begin{cases} 5a_0 + 11,25a_1 + 30,94a_2 = 11 \\ 11,25a_0 + 30,94a_1 + 94,92a_2 = 28,27 \\ 30,94a_0 + 94,92a_1 + 309,76a_2 = 88,14 \end{cases}$$

Решение этой системы  $a_0 = 4,54$ ;  $a_1 = -3,66$ ;  $a_2 = 0,95$ , и зависимость описывается формулой  $y = 4,54 - 3,66x + 0,95x^2$ .

**Погрешность приближения многочленом по методу наименьших квадратов:**

$$\Delta = \sqrt{\frac{1}{n+1} \sum_{i=0}^n (y_i - P_m(x_i))^2}$$

В случае нашего примера погрешность составит  $\varepsilon = 0,01$ .

Если экспериментальные точки располагаются вдоль некоторой линии, сходной по форме, например, с графиком гиперболической, показательной, логарифмической или других функций с неизвестными параметрами выбирается в качестве аппроксимирующей. Затем проводится *линеаризация* этой функции с помощью замены переменных и задача сводится к аппроксимации зависимости многочлена первой степени.

Например:

а). Показательная зависимость:  $y = ab^x$ , приводится к линейному виду путем логарифмирования  $\ln y = \ln a + x \ln b$ .

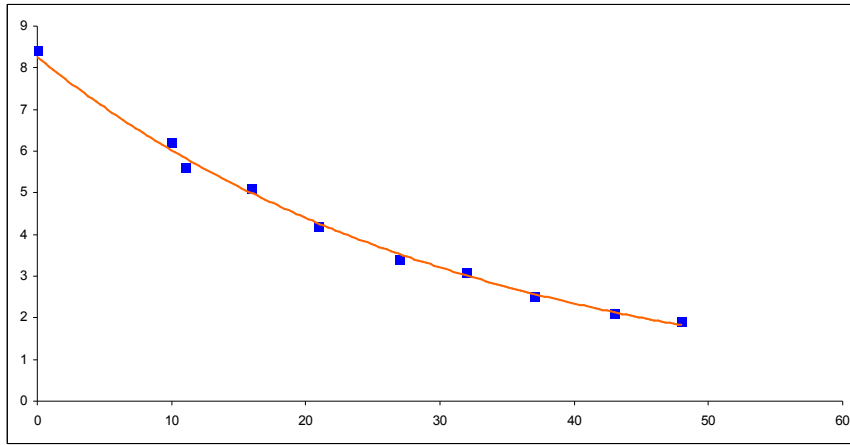
б) Степенная  $y = ax^b$ , аналогично  $\ln y = \ln a + b \ln x$ .

в) Гиперболическая  $y = \frac{x}{a + bx}$ , приводится к линейному виду введением новой переменной  $Y = x/y$ .

**Пример.** Результаты десяти наблюдений представлены в таблице. Установить вид зависимости между этими величинами и найти параметры эмпирической формулы.

$x$	0	10	11	16	21	27	32	37	43	48
$y$	8,4	6,2	5,6	5,1	4,2	3,4	3,1	2,5	2,1	1,9

Точечная диаграмма позволяет предположить, что зависимость между  $x$  и  $y$  экспоненциальная  $y = ae^{kx}$ . Вводим новые переменные  $Y = \ln y$ ,  $X = x$ ,  $b = \ln a$ , сводим зависимость к линейной  $Y = kX + b$ .



Перестраиваем табличную зависимость.

$x$	0	10	11	16	21	27	32	37	43	48
$\ln y$	2,128	1,825	1,723	1,629	1,435	1,824	1,131	0,916	0,71	0,642

По этим данным методом наименьших квадратов подберем аппроксимирующую линейную функцию.  $y = a_0 + a_1x$ . После решения системы получим  $a_0 = 2,11$ ,  $a_1 = -0,031 = k$ . Так как  $a_0 = b = \ln a$ , то  $a = e^{a_0} = e^{2,11} = 8,251$ .

Таким образом, полученная зависимость  $y = 8,251 \cdot e^{-0,031 \cdot x}$ .

### Численное интегрирование.

К вычислению определенного интеграла сводятся многие практические инженерные и научные задачи. Например, вычисление площадей фигур, объемов тел, работы силы и др.

**Постановка задачи:** Вычислить определенный интеграл

$$\int_a^b f(x) dx, \text{ где } f(x) \text{ – некоторая заданная на отрезке } [a, b] \text{ функция.}$$

Далеко не всегда можно вычислить интеграл в аналитическом виде по известной из курса математического анализа формуле Ньютона – Лейбница. Иногда даже при известной первообразной ее вид может оказаться очень сложным для вычислений. Также формулу Ньютона – Лейбница нельзя применить, если функция получена экспериментально в виде таблицы. Во всех этих случаях применяют численное интегрирование.

На практике наиболее широко используют квадратурные формулы:

$$\int_a^b f(x) dx \approx \sum_{i=0}^n A_i \cdot f(x_i)$$

где  $x_i$  – точки из отрезка  $[a, b]$  (узлы квадратурной формулы),  $A_i$  – числовые коэффициенты (веса) квадратурной формулы.

За приближенное значение интеграла принимаем  $\sum_{i=0}^n A_i \cdot f(x_i)$  - квадратурную сумму.

$$R = \int_a^b f(x) dx - \sum_{i=0}^n A_i \cdot f(x_i) \text{ - погрешность квадратурной формулы.}$$



Для вывода квадратурной формулы отрезок интегрирования  $[a, b]$  разбивается на  $n$  равных частичных отрезков  $[x_{i-1}, x_i]$ ,  $i = 1, \dots, n$  длиной  $h = (b - a) / n$ , а интеграл  $\int_a^b f(x) dx$

заменяется суммой частичных интегралов

$$\int_a^b f(x) dx \approx \sum_{i=0}^n \int_{x_{i-1}}^{x_i} f(x) dx.$$

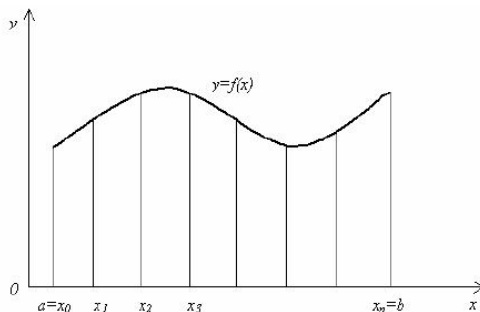
Затем подынтегральная функция  $f(x)$  на частичном отрезке  $[x_{i-1}, x_i]$  заменяется некоторым интерполяционным многочленом невысокой степени  $m$   $L_{m,i}(x)$ , и вычисляется интеграл  $\int_{x_{i-1}}^{x_i} f(x) dx = \int_{x_{i-1}}^{x_i} L_{m,i}(x) dx$  и приближенное значение интеграла будет получено в виде:

$$\int_a^b f(x) dx \approx \sum_{i=0}^n A_i \cdot f(x_i).$$

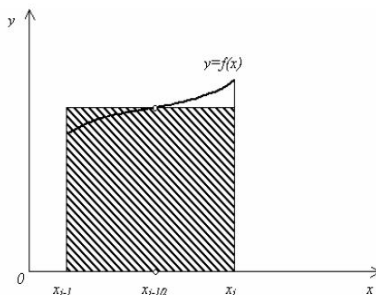
В зависимости от вида интерполяционного многочлена будут получаться различные квадратурные формулы. Если будет использоваться интерполяционный многочлен нулевой степени, то будет получена формула прямоугольников, если на отрезке  $[x_{i-1}, x_i]$  используется линейная интерполяция, то получается формула трапеций, если квадратичная – формула Симпсона. Но можно вывести простейшие квадратурные формулы исходя из геометрического смысла определенного интеграла, а именно находя приближенную площадь криволинейной трапеции.

$$S = \sum_{i=1}^n s_i = \sum_{i=1}^n f(x_i) \Delta x_i \quad \Delta x_i = x_i - x_{i-1}$$

### Формула прямоугольников.



Можно приближенно заменить площадь криволинейной трапеции площадью ступенчатой фигуры, состоящей из  $n$  прямоугольников. Площадь  $i$  – того прямоугольника находится как произведение длины отрезка основания  $[x_{i-1}, x_i]$  на значение функции в середине отрезка.  $s_i \approx h \cdot f(x_{i-1/2})$



Суммируя площади всех элементарных трапеций, получим **квадратурную формулу средних прямоугольников**.

$$\int_a^b f(x)dx \approx \sum_{i=1}^n f(x_{i-1/2})h = h(y_{1/2} + y_{3/2} + \dots + y_{n-3/2} + y_{n-1/2})$$

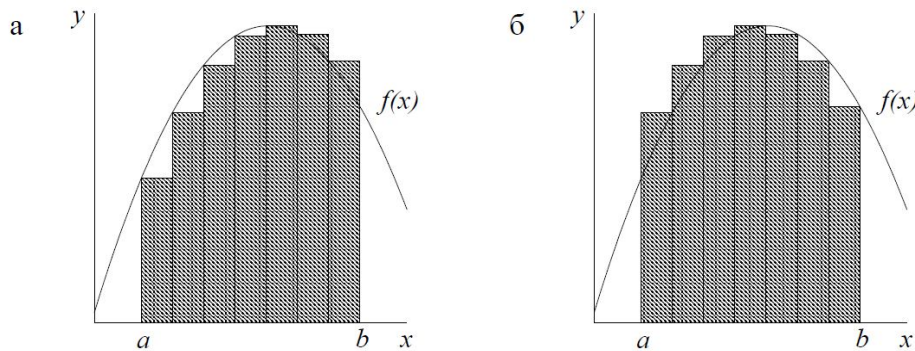
$$s_i \approx h \cdot f(x_i) \approx h \cdot f(x_{i-1}) \approx h \cdot f(x_{i-1/2})$$

Иногда используют формулы **правых и левых прямоугольников**:

$$\int_a^b f(x)dx \approx \sum_{i=1}^n f(x_i)h = h(y_1 + y_2 + \dots + y_{n-1} + y_n)$$

$$\int_a^b f(x)dx \approx \sum_{i=1}^n f(x_{i-1})h = h(y_0 + y_1 + \dots + y_{n-2} + y_{n-1})$$

Геометрическая иллюстрация этих формул.



### Формула трапеций.

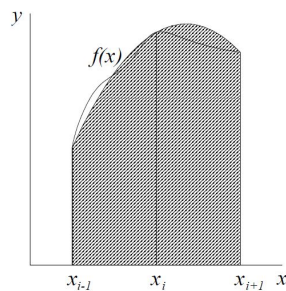
Формулу трапеций можно также получить из геометрических соображений. Отрезок интегрирования  $[a, b]$  разбивается на  $n$  частей длины  $h = \frac{b-a}{n}$ . ( $h$  - шаг разбиения.)

$x_1, x_2, \dots, x_n$  - абсциссы точек деления и соответствующие ординаты кривой  $y_1, y_2, \dots, y_n$  соединяются ломаной линией. В результате построения криволинейная трапеция разбивается на ряд вертикальных полосок одной и той же ширины  $h$ , каждую из которых приближенно можно принять за трапецию. Площадь элементарной трапеции:  $s_i = \frac{h}{2}(f(x_{i-1}) + f(x_i))$

Суммируя площади этих трапеций, получаем формулу:

$$\int_a^b f(x)dx \approx h \left( \frac{y_0}{2} + y_1 + y_2 + \dots + y_{n-1} + \frac{y_n}{2} \right) \text{ (Формула трапеций.)}$$

### Формула Симпсона.



Более точную формулу можно получить, если профиль криволинейной полоски считать параболой (используя многочлен Лагранжа второй степени), а не прямой линией как в формуле трапеций. В этом случае можно получить формулу Симпсона:

$$\int_a^b f(x)dx \approx \frac{b-a}{3n} \{f(a) + 2[f(a+2h) + \dots + f(a+(n-2)h)] + \\ + 4[f(a+h) + \dots + f(a+(n-1)h)] + f(b)\}, \quad h = \frac{b-a}{n}.$$

Для практической оценки погрешности интеграла используется правило Рунге. Для этого проводят вычисления с шагом  $h$  и  $h/2$  и получают приближенные значения интегралов  $I_h$  и  $I_{h/2}$ .

За погрешность приближенного значения интеграла принимают величину:  $(I_{h/2} - I_h)/3$  - для формулы прямоугольников и трапеций,  $(I_{h/2} - I_h)/15$  - по формуле Симпсона.

В адаптивных алгоритмах автоматически определяют величину шага  $h$ , таким образом, чтобы результат удовлетворял заданной точности. За окончательные значения интеграла принимают:  $I_{h/2} + (I_{h/2} - I_h)/3$  - по формуле прямоугольников,  $I_{h/2} - (I_{h/2} - I_h)/3$  - формуле трапеций,  $I_{h/2} - (I_{h/2} - I_h)/15$  - формуле Симпсона.

### Решение обыкновенных дифференциальных уравнений.

Инженеру часто приходится иметь дело с техническими системами и технологическими процессами, характеристики которых непрерывно меняются со временем  $t$ . Эти явления, как правило, подчиняются физическим законам, которые формулируются в виде дифференциальных уравнений. Например, кинетика химических реакций, динамика биологических популяций, движение космических объектов, модели экономического развития, все эти процессы исследуются с помощью обыкновенных дифференциальных уравнений (ОДУ).

*Обыкновенным дифференциальным уравнением первого порядка* называется уравнение вида  $F(x, y, y') = 0$  или  $y' = f(x, y)$ . Функция  $y(x)$ , при подстановке которой уравнение обращается в тождество, называется *решением дифференциального уравнения*.

Для решения обыкновенных дифференциальных уравнений могут применяться точные методы решения, а также приближенные (аналитические) методы, такие, например, как разложение в числовой ряд, рассмотренные в курсе высшей математики. К приближенным методам решения относятся и численные методы.

При решении ДУ численным методом приближенные значения искомой функции  $y(x)$  находят для ряда значений  $x$  из отрезка, на котором нужно найти решение. С помощью численных методов невозможно найти общее решение, а только частное решение, то есть решение *задачи Коши*. Подавляющее большинство возникающих на практике задач Коши невозможно решить без использования вычислительной техники. Поэтому в инженерных и научно-технических расчетах численные методы решения задачи Коши играют важную роль.

Рассмотрим несколько численных методов решения дифференциальных уравнений первого порядка.

**Постановка задачи:** Решить уравнение  $y' = f(x, y)$ , при начальном условии  $y(x_0) = y_0$ , где  $x, x_0 \in [a, b]$ .

Первый этап на пути построения численных методов решение на отрезке  $[a, b]$  задачи Коши состоит в замене отрезка  $[a, b]$  – области непрерывного изменения аргумента  $x$  – множеством конечного числа точек  $a = x_0 < x_1 < \dots < x_i < \dots < x_n = b$ , называемого *сеткой*. Самы точки  $x_i$  - *узлы сетки*. Величина  $h = x_i - x_{i-1}$  - *шаг сетки*. Будем рассматривать сетки, в

которых шаг  $h$  - постоянный. В таком случае  $x_i = a + i h$ ,  $h = (b-a)/n$ , и сетка называется *равномерной*.

Следующий этап в построении численного метода состоит в замене задачи Коши  $y' = f(x, y)$ ,  $y(x_0) = y_0$  ее дискретным аналогом – решая которую можно последовательно найти таблицу приближенных значений  $y_0, y_1, \dots, y_i, \dots, y_n$ , решения  $y(x)$  в узлах сетки, то есть  $y(x_i) = y_i$ .

Численный метод решения задачи Коши называется *одношаговым*, если для вычисления решения в точке  $x_i + h$  используется информация о решении только в точке  $x_i$ .

Простейший одношаговый метод численного решения задачи Коши – метод Эйлера.

### Метод Эйлера.

Метод Эйлера является сравнительно грубым и применяется в основном для ориентировочных расчетов. Однако идеи, положенные в основу метода Эйлера, являются исходными для ряда других методов.

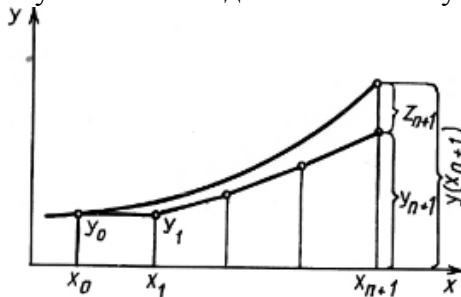
Идея метода заключается в том, что на малом промежутке изменения независимой переменной интегральная кривая дифференциального уравнения заменяется отрезком прямой (касательной).

Пусть необходимо найти решение уравнения  $y' = f(x, y)$  (1) с начальным условием  $y(x_0) = y_0$  (2). Разложим искомую функцию  $y(x)$  в ряд вблизи точки  $x_0$  и ограничимся первыми двумя членами разложения  $y(x) = y(x_0) + y'(x_0)(x - x_0) + \dots$ . Учитываем уравнение (1) и обозначив  $h = x - x_0$ , получаем  $y(x) \approx y(x_0) + f(x_0, y_0) \cdot h$ . Эту формулу можно применять многократно, находя значения функции во все новых и новых точках.

$$y_{i+1} = y_i + h f(x_i, y_i) \quad (i=0, 1, 2, \dots). \quad (3)$$

Такой метод решения обыкновенных дифференциальных уравнений называется *методом Эйлера*.

Запускается метод из начальных условий  $y(x_0) = y_0$ .



Геометрически метод Эйлера означает, что на каждом шаге мы аппроксимируем решение (интегральную кривую) отрезком касательной, проведенной к графику решения в начале интервала. При этом искомая интегральная кривая  $y = y(x)$ , проходящая через точку  $M_0(x_0, y_0)$ , заменяется ломаной  $M_0M_1M_2 \dots$  с вершинами  $M_i(x_i, y_i)$  ( $i=0, 1, 2, \dots$ ); каждое звено  $M_iM_{i+1}$  этой ломаной, называемой *ломаной Эйлера*, имеет направление, совпадающее с направлением той интегральной кривой уравнения, которая проходит через точку  $M_i$ .

Метод Эйлера имеет низкую точность и для того чтобы получить приемлемую погрешность необходимо делать шаг расчета достаточно маленьким. Можно улучшить точность метода Эйлера, улучшив аппроксимацию производной. Это можно сделать, например, используя среднее значение производной в начале и в конце интервала.

### Модифицированный метод Эйлера.

Рассматриваем дифференциальное уравнение  $y' = f(x, y)$  с начальным условием  $y(x_0) = y_0$ .

Вычислим значение функции в следующей точке по методу Эйлера:  $y_{i+1}^* = y_i + h f(x_i, y_i)$  и используем это значение для нахождения производной в конце интервала  $f(x_{i+1}, y_{i+1}^*)$ . Вместо значения  $f(x_i, y_i)$  в формуле Эйлера используем полусумму  $f(x_i, y_i)$  и  $f(x_{i+1}, y_{i+1}^*)$ .

$$y_{i+1} = y_i + h \cdot \frac{1}{2} \cdot (f(x_i, y_i) + f(x_{i+1}, y_{i+1}^*)). \quad (4)$$

Разложим функцию в ряд Тейлора вблизи точки  $x_0$  и добавим еще один член разложения:  $y(x) = y(x_0) + y'(x_0)(x - x_0) + \frac{1}{2} y''(x_0)(x - x_0)^2 + \dots$ . Заменим вторую производную разностным уравнением  $y''(x_0) = \frac{y'(x_0 + h) - y'(x_0)}{h}$  и подставим в разложение.

$$y(x + h) \approx y(x_0) + y'(x_0) \cdot h + \frac{1}{2} (y'(x_0 + h) - y'(x_0)) / h \cdot h^2$$

Преобразуем

$$y(x + h) \approx y(x_0) + y'(x_0) \cdot h + \frac{1}{2} (y'(x_0 + h) - y'(x_0)) \cdot h$$

$$y(x + h) \approx y(x_0) + \frac{1}{2} (y'(x_0 + h) + y'(x_0)) \cdot h$$

Эта формула совпадает с уже полученным выражением (4). При этом точное значение  $y'(x_0 + h)$  заменено приближенным  $f(x_{i+1}, y_{i+1}^*)$ . При вычислении модифицированным методом Эйлера, сначала вычисляют значение  $y_{i+1}$ , а затем уже подставляют его в формулу (4).

### Метод Рунге – Кутты.

Методом Рунге – Кутты четвертого порядка точности называют одношаговый метод, относящийся к широкому классу методов Рунге-Кутты. В этом методе величины  $y_{i+1}$  вычисляются по следующим формулам:

$$y_{i+1} = y_k + (m_1 + 2m_2 + 2m_3 + m_4) \cdot \frac{h}{6},$$

где

$$m_1 = f(x_i, y_i)$$

$$m_2 = f\left(x_i + \frac{h}{2}, y_i + m_1 \cdot \frac{h}{2}\right)$$

$$m_3 = f\left(x_i + \frac{h}{2}, y_i + m_2 \cdot \frac{h}{2}\right)$$

$$m_4 = f\left(x_i + \frac{h}{2}, y_i + m_3 \cdot h\right).$$

**Пример.** Решить дифференциальное уравнение  $y' = y + 3x$ , при начальном условии  $y(0) = -1$ , где  $x \in [0; 0,5]$ .

Точное аналитическое решение задачи Коши:  $y = 2e^x - 3x - 3$ .

Решим методом Эйлера.

Выберем для решения шаг  $h = 0,1$ .

Формула для расчета  $y_{i+1} = y_i + h \cdot f(x_i, y_i)$ , т.е.  $y_{i+1} = y_i + h \cdot (3x_i + y_i)$ ,  $x_i = 0,1 \cdot i$

Чтобы получить решение в точке  $x_1$ , используем значения в начальной точке.

$$y_1 = y_0 + h \cdot (3x_0 + y_0), \quad y_1 = -1 + 0,1 \cdot (3 \cdot 0 - 1) = -1,1, \quad x_1 = 0,1 \cdot 1 = 0,1.$$

Следующая точка:  $x_2 = 0,1 \cdot 2 = 0,2$

$$y_2 = y_1 + h \cdot (3x_1 + y_1), \quad y_2 = -1,1 + 0,1 \cdot (3 \cdot 0,1 - 1,1) = -1,18$$

Запишем результаты расчетов в таблицу.

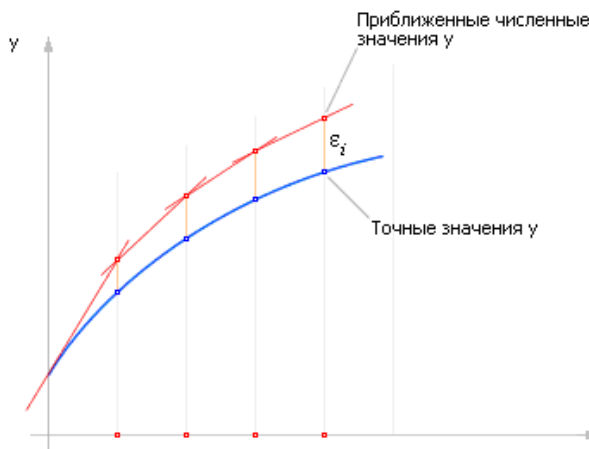
$i$	$x$	Точное решение	$h=0.1$	Относительная погрешность
0	0	-1	-1	0
1	0,1	-1,08966	-1,1	0,940167
2	0,2	-1,15719	-1,18	1,932671
3	0,3	-1,20028	-1,238	3,046657
4	0,4	-1,21635	-1,2718	4,359915
5	0,5	-1,20256	-1,27898	5,975273

Проведем компьютерный расчет с шагами  $h = 0,01$  и  $h = 0,001$ . Результаты показаны в таблице.

x	Точное решение	$h=0,01$	Относительная погрешность	$h=0,001$	Относительная погрешность
0	-1	-1	0	-1	0
0,1	-1,08966	-1,09076	0,100728	-1,08977	0,010135
0,2	-1,15719	-1,15962	0,209596	-1,15744	0,021095
0,3	-1,20028	-1,2043	0,334903	-1,20069	0,033714
0,4	-1,21635	-1,22227	0,48686	-1,21695	0,049022
0,5	-1,20256	-1,21074	0,680125	-1,20338	0,068496

### Оценка погрешности по правилу Рунге.

Численное решение дифференциального уравнения всегда получается с погрешностью.



Практически оценить погрешность метода позволяет правило Рунге. Сначала вычисляют приближенное решение с шагом  $h$ , затем - с шагом  $h/2$ . и сравнивают величины  $y_i^{(h)}$  и  $y_i^{(h/2)}$ . Оценкой погрешности метода будет величина:  $\frac{\max|y_i^h - y_i^{h/2}|}{2^p - 1}$ , где  $p$  – порядок точности метода.

Для метода Эйлера ( $p = 1$ ):  $\max|y_i^h - y_i^{h/2}|$

Для модифицированного метода Эйлера ( $p = 2$ ):  $\frac{\max|y_i^h - y_i^{h/2}|}{3}$

Для метода Рунге-Кутты ( $p = 4$ ):  $\frac{\max|y_i^h - y_i^{h/2}|}{15}$ .